Sample project on Jenkins CICD with GitHub Integration:

* Push the demo project into GitHub repository.
* Run the Ec2 instance (ubuntu).
* Install Java and Jenkins using below commands.
* sudo apt update
* sudo apt install openjdk-11-jre
* java -version
* curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \   /usr/share/keyrings/jenkins-keyring.asc > /dev/null
* echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \   https://pkg.jenkins.io/debian binary/ | sudo tee \   /etc/apt/sources.list.d/jenkins.list > /dev/null
* sudo apt-get update.
* sudo apt-get install jenkins
* sudo systemctl enable jenkins
* sudo systemctl start jenkins
* sudo systemctl status Jenkins
* Once the Jenkins are up and running, in order to access we need to create the inbound security rule on 8080 port.
* Public Ip 8080
* Create a job in Jenkins.
* Integrate the GitHub and Jenkins by by providing the credentials.
* Build run.
* Will get the console output.
* We need to check whether the Jenkins integrated with GitHub or not.
* After integration we need to execute the commands like:
* sudo cat /var/lib/Jenkins/secrets/initialAdminPassword.

Sudo apt install node.js

Sudo apt install npm

Sudo npm install.

Node app.js

* We need to create the inbound rule on 8000 port because to give public access.
* Then the application will be executed, If we exit in instance then the application will be disappeared .
* So in order to resolve that we need run or integrate our project in docker.
* We need to install docker and create the docker file and we need to execute the below commands.
* sudo apt install docker.io
* FROM node:12.2.0-alpine
* WORKDIR app
* COPY.
* RUN npm install.
* EXPOSE 8000
* CMD ["node","app.js"]
* docker build. -t node-app
* sudo usermod -a -G docker $USER
* docker run -d --name node-todo-app -p 8000:8000 todo-node-app.
* After that we need do,

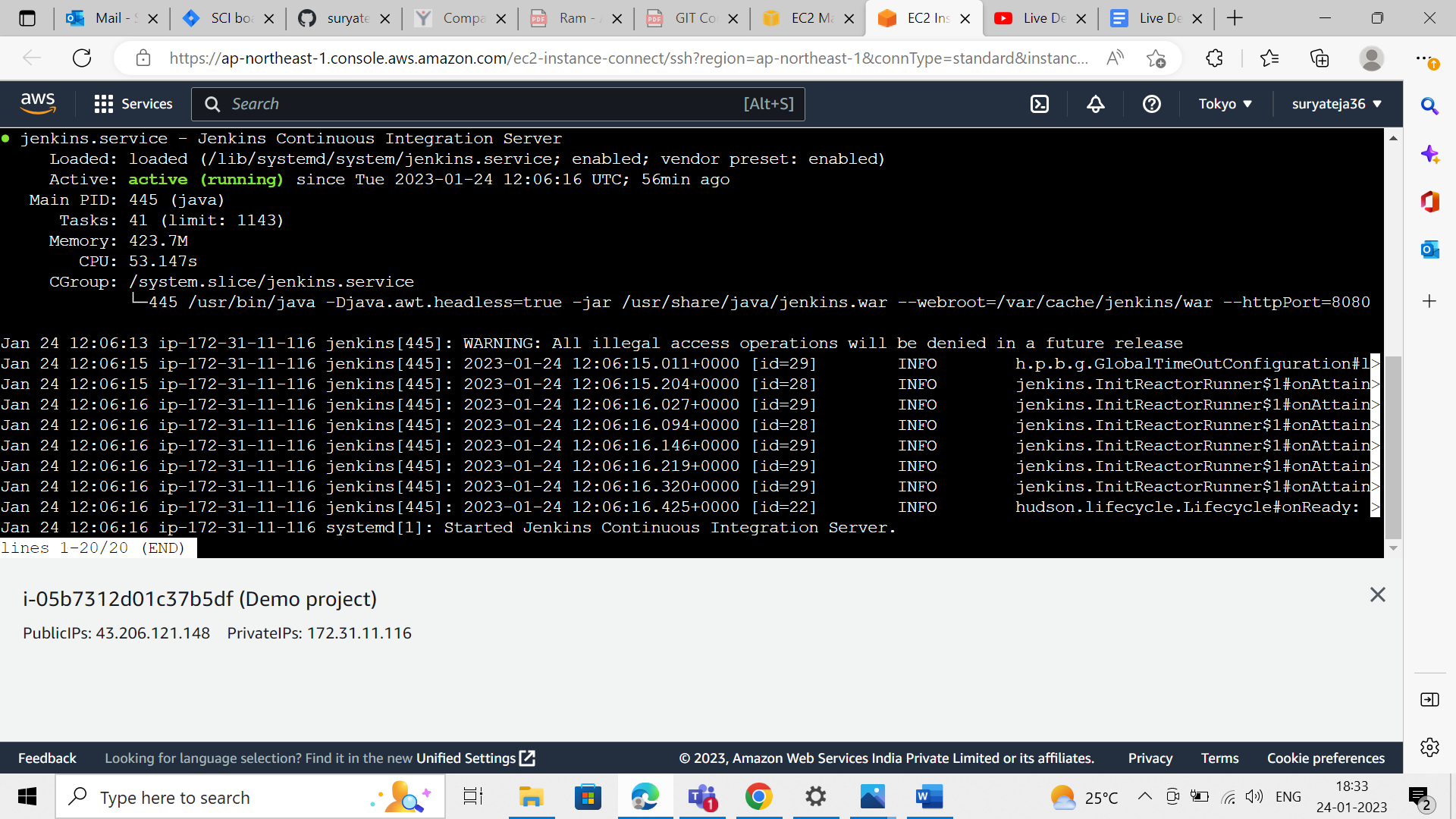
Got to jenkins job.

Execute shell.

docker build. -t node-app-todo

docker run -d --name node-app-container -p 8000:8000 node-app-todo

* We need to build the code then we get the console output.



A screenshot of a computer

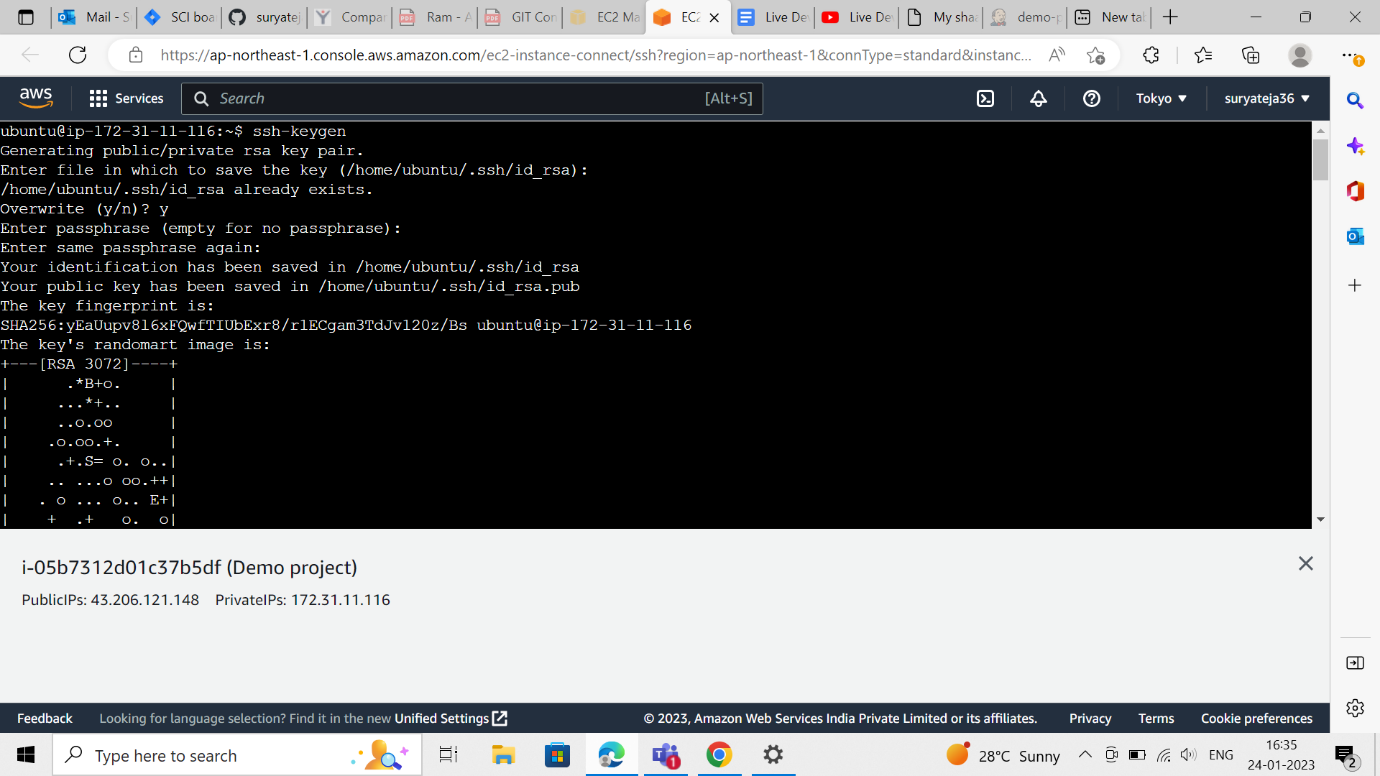
Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



A screenshot of a computer

Description automatically generated

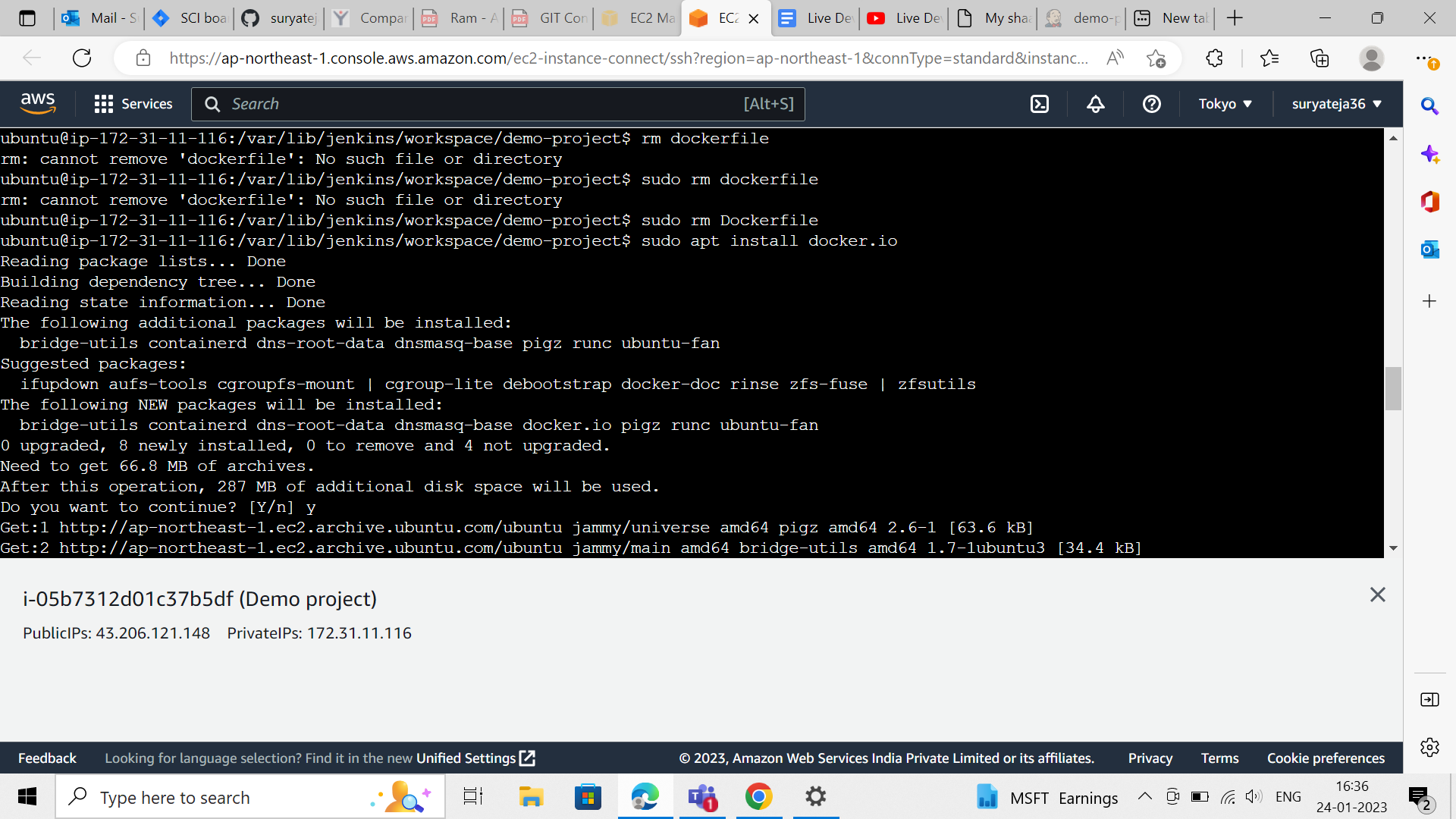
A screenshot of a computer

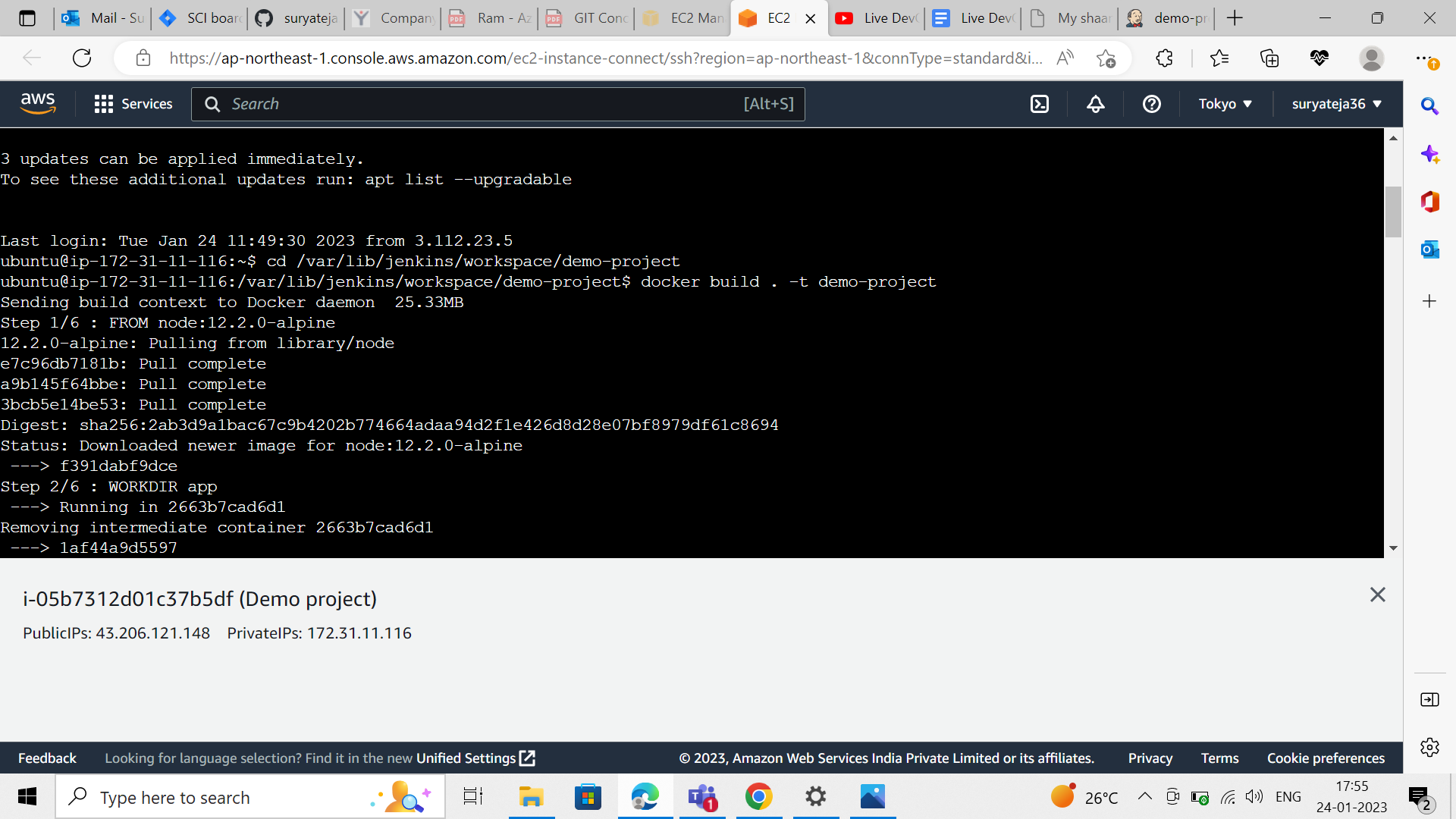
Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

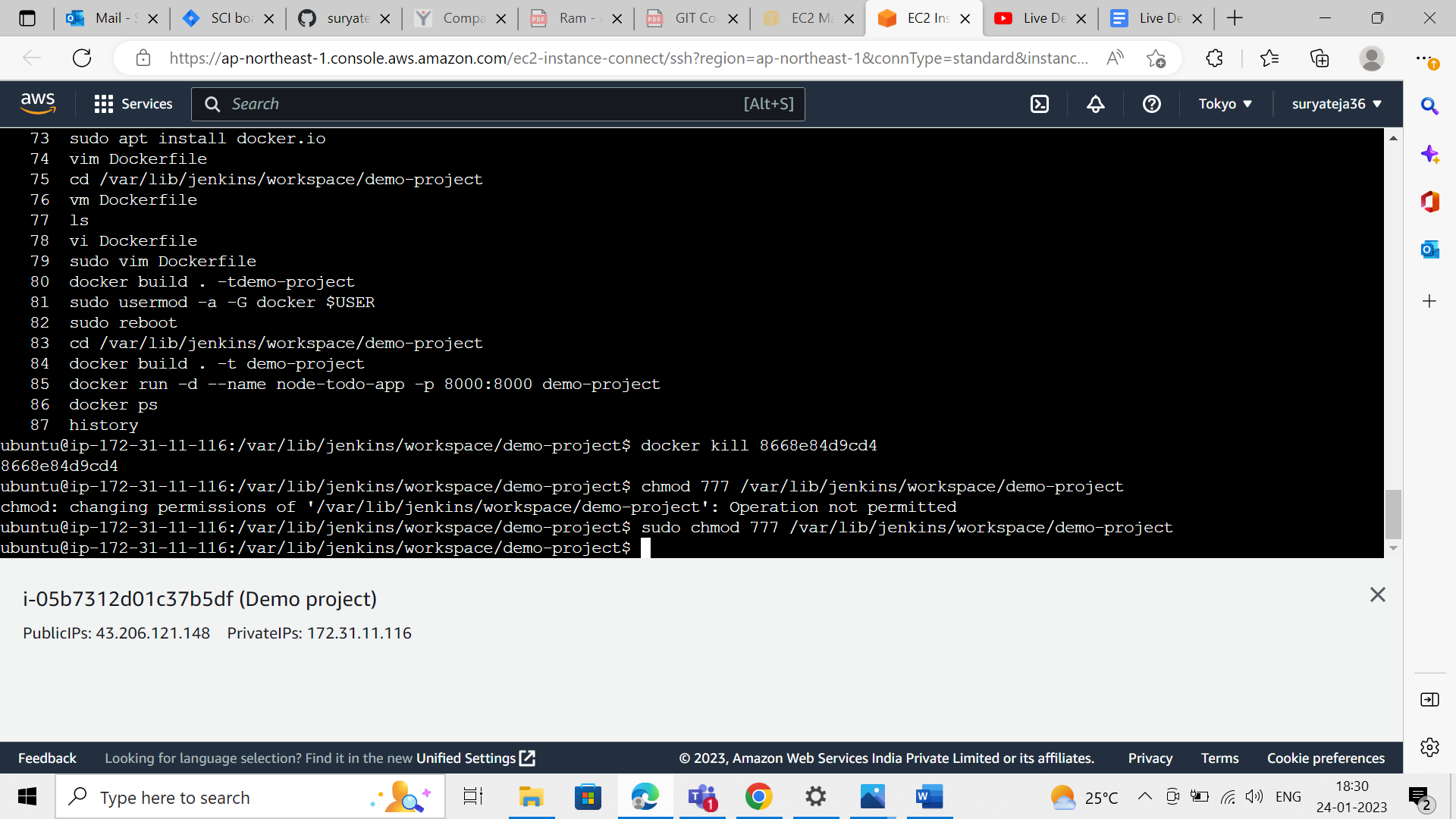
* Docker commands execution





A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated